

Carousel

The Carousel component is a modern, interactive image gallery component that provides an engaging way to display collections of images or content. Built with accessibility in mind, it features touch/swipe support, keyboard navigation, and intelligent card positioning that creates a stacked, 3D-like effect. The `<aava-carousel>` component automatically handles responsive behavior, touch interactions, and provides smooth transitions between items while maintaining excellent performance through smart rendering strategies.

How to use

```
■ import { AavaCarouselComponent, CarousellItem } from "@aava/play-core";
```

Explore different carousel configurations and sizes with interactive demos.

Basic Usage

```
■ Angular Preview Code < aava-carousel [items] = " carousellItems " > </ aava-carousel > carousellItems : CarousellItem [ ] = [ { id : '1' , imageUrl : 'assets/center.jpg' , imageAlt : 'Image 1' , routeUrl : '' , } , { id : '2' , imageUrl : 'assets/right-1.jpg' , imageAlt : 'Image 2' , routeUrl : '' , } , { id : '3' , imageUrl : 'assets/right-2.jpg' , imageAlt : 'Image 3' , routeUrl : '' , } , { id : '4' , imageUrl : 'assets/left-2.jpg' , imageAlt : 'Image 4' , routeUrl : '' , } , { id : '5' , imageUrl : 'assets/left-1.jpg' , imageAlt : 'Image 5' , routeUrl : '' , } , ] ;
```

The basic carousel demonstrates the default configuration with a set of items. It shows how items are displayed in a 3D stacked layout with smooth transitions. This is a great starting point for integrating a carousel in your application.

Sizes

```
■ Angular Preview Code < aava-carousel [items] = " carousellItems " size = " xs " > </ aava-carousel > < aava-carousel [items] = " carousellItems " size = " sm " > </ aava-carousel > < aava-carousel [items] = " carousellItems " size = " md " > </ aava-carousel > < aava-carousel [items] = " carousellItems " size = " lg " > </ aava-carousel >
```

carousellItems : CarousellItem [] = [{ id : '1' , imageUrl : 'assets/center.jpg' , imageAlt : 'Image 1' , routeUrl : '' , } , { id : '2' , imageUrl : 'assets/right-1.jpg' , imageAlt : 'Image 2' , routeUrl : '' , } , { id : '3' , imageUrl : 'assets/right-2.jpg' , imageAlt : 'Image 3' , routeUrl : '' , } , { id : '4' , imageUrl : 'assets/left-2.jpg' , imageAlt : 'Image 4' , routeUrl : '' , } , { id : '5' , imageUrl : 'assets/left-1.jpg' , imageAlt : 'Image 5' , routeUrl : '' , } ,] ;

Available Sizes

Carousels come in four size variants to fit different interface needs. The size affects the card dimensions and spacing. Use smaller sizes for compact layouts and larger sizes for prominent sections.

- xs (Extra Small) - Extra compact size for very dense interfaces
- sm (Small) - Compact size for dense interfaces
- md (Medium) - Standard size for most use cases (default)
- lg (Large) - Prominent size for primary actions

Autoplay

```
■ Angular Preview Code < aava-carousel [items] = " carousellItems " [autoplay] = " true " > </ aava-carousel > carousellItems : CarousellItem [ ] = [ { id : '1' , imageUrl : 'assets/center.jpg' , imageAlt : 'Image 1' , routeUrl : '' , } , { id : '2' , imageUrl : 'assets/right-1.jpg' , imageAlt : 'Image 2' , routeUrl : '' , } , { id : '3' , imageUrl : 'assets/right-2.jpg' , imageAlt : 'Image 3' , routeUrl : '' , } , { id : '4' , imageUrl : 'assets/left-2.jpg' , imageAlt : 'Image 4' , routeUrl : '' , } , { id : '5' , imageUrl : 'assets/left-1.jpg' , imageAlt : 'Image 5' , routeUrl : '' , } , ] ;
```

■ The autoplay feature automatically transitions slides at the specified interval. It's ideal for galleries or highlight sections where user interaction may be minimal.

With Titles

```
■ Angular Preview Code Carousels can also display item
```

titles for additional context or labeling. This variant is useful when each slide represents a category, product, or feature that requires a clear textual description.

Carousel Features

- **Touch & Swipe Support** : Full mobile touch interaction with swipe gestures
- **Mouse Navigation** : Drag and drop functionality with wheel support
- **Keyboard Access** : Arrow keys, home, and end key
- **Visual Indicators** : Navigation dots and arrow controls
- **Size Variants** : Four size options (xs, sm, md, lg) for different use cases
- **Smart Positioning** : Intelligent card stacking and positioning
- **Accessibility** : ARIA support and screen reader compatibility
- **Features**

- The carousel provides comprehensive touch interaction:
- **Swipe Navigation** : Left/right swipe to navigate between items
- **Touch Sensitivity** : Configurable swipe threshold (default: 50px)
- **Multi-touch Support** : Handles single-finger gestures properly
- **Passive Events** : Optimized touch event handling for performance
- **Mouse & Keyboard Navigation**

 - Multiple input methods for desktop users:
 - **Mouse Drag** : Click and drag to navigate through items
 - **Wheel Support** : Trackpad two-finger swipe support
 - **Keyboard Shortcuts** : Arrow Left/Right : Navigate between items
 - **Home** : Jump to first item
 - **End** : Jump to last item
 - **Focus Management** : Proper focus trapping and restoration

- **Smart Card Positioning**

 - Advanced positioning system for visual appeal:
 - **Stacked Layout** : Cards are positioned in 3D space with depth
 - **Dynamic Scaling** : Cards scale based on distance from center
 - **Optimal Visibility** : Shows maximum number of cards without overlap
 - **Smooth Transitions** : CSS transitions for fluid animations
 - **Responsive Behavior** : Adapts positioning based on screen size

API Reference

- **Input Properties**
- **Property Type**
- **Description**

CarousellItem[] [] Array of carousel items to display

- **activeIndex** number 0 Initial active item index
- **size** 'lg' | 'md' | 'sm' | 'xs' 'md' Size variant for the carousel
- **customStyles** Record<string, string> {} Inline CSS custom properties to override defaults
- **autoplay** boolean false Enables automatic slide transition
- **autoplayInterval** number 2000 Interval duration (in ms) for autoplay transitions

Events

- **Event Type** Description
- **itemChange** EventEmitter<{ item: CarousellItem; index: number }> Emitted whenever the active item changes

Methods

- **Navigation Methods**
- **goToPrevious ()** : void
- **goToNext ()** : void
- **goToSlide (index : number)** : void
- **toggleAutoplay ()** : void
- **startAutoplay ()** : void
- **stopAutoplay ()** : void
- **goToPrevious()** — Move to the previous carousel item.
- **goToNext()** — Move to the next carousel item.
- **goToSlide(index)** — Jump directly to a specific slide.
- **toggleAutoplay()** — Toggles autoplay on or off.
- **startAutoplay()** — Starts autoplay manually.
- **stopAutoplay()** — Stops autoplay manually.

CSS Custom Properties

- **Variable Description**
- **--carousel-in-duration** Duration for carousel entrance animation
- **--carousel-out-duration** Duration for carousel exit animation
- **--carousel-container-width** Width of the carousel container
- **--carousel-container-display** Display property for the carousel container
- **--carousel-container-flex-direction** Flex direction of the carousel layout
- **--carousel-container-align-items** Alignment of items inside the container
- **--carousel-viewport-height-xs** Viewport height for extra small screens
- **--carousel-viewport-height-sm** Viewport height for small screens
- **--carousel-viewport-height-md** Viewport height for medium screens
- **--carousel-viewport-height-lg** Viewport height for large screens
- **--carousel-radius-xs** Border radius for extra small size
- **--carousel-radius-sm** Border radius for small size
- **--carousel-radius-md** Border radius for medium size
- **--carousel-radius-lg** Border

radius for large size --carousel-image-radius Border radius for carousel images --carousel-bg
Background color of the carousel --carousel-shadow Default shadow styling for carousel cards
--carousel-shadow-hover Shadow styling on hover --carousel-transition-duration Transition
duration for carousel animations --carousel-transition-timing Timing function for transitions
--carousel-shadow-transition Transition property for shadow changes --carousel-z-center Z-index
for the center card --carousel-z-side-1 Z-index for first side cards --carousel-z-side-2 Z-index for
second side cards --carousel-z-hidden Z-index for hidden cards --carousel-opacity-center Opacity
for center card --carousel-opacity-side-1 Opacity for first side cards --carousel-opacity-side-2
Opacity for second side cards --carousel-opacity-hidden Opacity for hidden cards
--carousel-scale-center Scale for center card --carousel-scale-side-1 Scale for first side cards
--carousel-scale-side-2 Scale for second side cards --carousel-scale-hidden Scale for hidden
cards --carousel-center-width-md Width of the center card for medium screens
--carousel-center-height-md Height of the center card for medium screens
--carousel-side-1-translate-md Translate distance for first side cards (medium)
--carousel-side-2-translate-md Translate distance for second side cards (medium)
--carousel-controls-margin-top Top margin for carousel controls --carousel-controls-gap Gap
between carousel controls --carousel-indicators-gap Gap between carousel indicators
--carousel-indicator-size Size of the indicator dot --carousel-indicator-active-size Size of the active
indicator --carousel-indicator-color Default indicator color --carousel-indicator-active-color Color of
the active indicator --carousel-indicator-border-radius Border radius for indicator dots
--carousel-icon-color Color of navigation icons --carousel-icon-size Size of navigation icons
--carousel-icon-transition Transition for icon color changes Best Practices ■ Content Organization
■ Image Quality : Use appropriately sized images for each carousel size Alt Text : Always provide
descriptive alt text for accessibility Item Count : Limit to 5-10 items for optimal performance
Consistent Sizing : Use images with similar aspect ratios User Experience ■ Touch Targets :
Ensure navigation elements are large enough for touch Visual Feedback : Provide clear indication
of current active item Loading States : Handle image loading gracefully Error Handling : Provide
fallbacks for failed image loads Performance ■ Image Optimization : Use compressed images and
appropriate formats Lazy Loading : Consider implementing lazy loading for large galleries Change
Detection : Component uses OnPush strategy for optimal performance Memory Management :
Proper cleanup of event listeners Accessibility ■ Keyboard Navigation : Ensure all functionality is
keyboard accessible Screen Readers : Provide meaningful alt text and ARIA labels Focus
Management : Maintain proper focus order and visibility High Contrast : Ensure sufficient contrast
for all text and controls

```
<aava-carousel [items]="carouselItems"> </aava-carousel>
```

```
---
```

```
carouselItems: CarouselItem[] = [
  {
    id: '1',
    imageUrl: 'assets/center.jpg',
    imageAlt: 'Image 1',
    routeUrl: '',
  },
  {
    id: '2',
    imageUrl: 'assets/right-1.jpg',
    imageAlt: 'Image 2',
    routeUrl: '',
  },
  {
    id: '3',
    imageUrl: 'assets/right-2.jpg',
    imageAlt: 'Image 3',
    routeUrl: '',
  },
  {
    id: '4',
    imageUrl: 'assets/left-2.jpg',
    imageAlt: 'Image 4',
    routeUrl: '',
  },
  {
    id: '5',
    imageUrl: 'assets/left-1.jpg',
    imageAlt: 'Image 5',
    routeUrl: '',
  },
];
```

```
<aava-carousel [items]="carouselItems" size="xs"> </aava-carousel>
<aava-carousel [items]="carouselItems" size="sm"> </aava-carousel>
<aava-carousel [items]="carouselItems" size="md"> </aava-carousel>
<aava-carousel [items]="carouselItems" size="lg"> </aava-carousel>
```

```
carouselItems: CarouselItem[] = [
  {
    id: '1',
    imageUrl: 'assets/center.jpg',
    imageAlt: 'Image 1',
    routeUrl: '',
  },
  {
    id: '2',
    imageUrl: 'assets/right-1.jpg',
    imageAlt: 'Image 2',
    routeUrl: '',
  },
  {
    id: '3',
    imageUrl: 'assets/right-2.jpg',
    imageAlt: 'Image 3',
    routeUrl: '',
  },
  {
    id: '4',
    imageUrl: 'assets/left-2.jpg',
    imageAlt: 'Image 4',
    routeUrl: '',
  },
  {
    id: '5',
    imageUrl: 'assets/left-1.jpg',
    imageAlt: 'Image 5',
    routeUrl: '',
  },
];
```

```
<aava-carousel [items]="carouselItems" [autoplay]="true"> </aava-carousel>

---

carouselItems: CarouselItem[] = [
  {
    id: '1',
    imageUrl: 'assets/center.jpg',
    imageAlt: 'Image 1',
    routeUrl: '',
  },
  {
    id: '2',
    imageUrl: 'assets/right-1.jpg',
    imageAlt: 'Image 2',
    routeUrl: '',
  },
  {
    id: '3',
    imageUrl: 'assets/right-2.jpg',
    imageAlt: 'Image 3',
    routeUrl: '',
  },
  {
    id: '4',
    imageUrl: 'assets/left-2.jpg',
    imageAlt: 'Image 4',
    routeUrl: '',
  },
  {
    id: '5',
    imageUrl: 'assets/left-1.jpg',
    imageAlt: 'Image 5',
    routeUrl: '',
  },
];
```

■ No code found